# Artificial Neural Network-Aided Image Analysis System for Cell Counting

**Per Jesper Sjöström,**[1] **Beata Ras Frydel,**[1] **and Lars Ulrik Wahlberg**[1,2]*

[1]CytoTherapeutics, Inc., Lincoln, Rhode Island

[2]Department of Clinical Neuroscience, Section of Neurosurgery, Karolinska Institute, Stockholm, Sweden

Received 29 May 1998; Revision Received 11 December 1998; Accepted 15 January 1999

**Background:** In histological preparations containing debris and synthetic materials, it is difficult to automate cell counting using standard image analysis tools, i.e., systems that rely on boundary contours, histogram thresholding, etc. In an attempt to mimic manual cell recognition, an automated cell counter was constructed using a combination of artificial intelligence and standard image analysis methods.

**Methods:** Artificial neural network (ANN) methods were applied on digitized microscopy fields without pre-ANN feature extraction. A three-layer feed-forward network with extensive weight sharing in the first hidden layer was employed and trained on 1,830 examples using the error back-propagation algorithm on a Power Macintosh 7300/180 desktop computer. The optimal number of hidden neurons was determined and the trained system was validated by comparison with blinded human counts. System performance at 50x and 100x magnification was evaluated.

**Results:** The correlation index at 100x magnification neared person-to-person variability, while 50x magnification was not useful. The system was approximately six times faster than an experienced human.

**Conclusions:** ANN-based automated cell counting in noisy histological preparations is feasible. Consistent histology and computer power are crucial for system performance. The system provides several benefits, such as speed of analysis and consistency, and frees up personnel for other tasks. Cytometry 36:18–26, 1999. © 1999 Wiley-Liss, Inc.

**Key terms:** cell count; neural networks (computer); back-propagation; artificial intelligence; image processing; computer-assisted; NIH Image; microscopy; encapsulation

Cell counting is often a tedious task, for which error-prone and impatient humans are not particularly well suited. It is sometimes possible to automate this process using traditional ‘‘off-the-shelf’’ image analysis tools (systems that rely on textures, boundary contours, histogram thresholding, gray-scale intensity peak search, etc.). However, for our purposes, we found it did not work well. Our aim was to automate cell counting in encapsulated devices that are used for central nervous system drug delivery (1–3). Problems arose due to the variable physiognomy of cells, background noise in the form of cellular and biomaterial debris, histological variations, and the presence of other cell types. Manual cell counting was, therefore, needed but was time-consuming and often gave inconsistent results—cell numbers varied between times of observations and between observers. In order to automate the process, some of the intelligence of the human being should ideally be combined with the consistency and speed of the computer. This may be accomplished with artificial neural networks (ANN) that can mimic human expertise through learning by example.

An ANN is a parallel, computer-based data processing system inspired by the brain, and consists of simple artificial neurons. These can be organized in a sequence of layers without any recurrent connections in a structure called a feed-forward ANN. This type of ANN is suitable for classification tasks and was used in this study. The neural interconnections are represented by synaptic weights. The data is fed to the input layer by weighted summation and is transferred to the succeeding layer through a limiting activation function. The process is repeated for the subsequent layers, eventually producing an output answer. The result of a neuron in the output layer can be chosen to signify the class to which the input data belongs and the ANN can learn to provide a desired output for a given input sample. The training of an ANN involves gradually adjusting the synaptic weights according to a back-propagated output error for each sample, until the desired responses are on average obtained on the entire

Grant sponsor: CytoTherapeutics, Inc.

Per Jesper Sjöström’s present address is Brandeis University, Biology Department, 415 South Street, Waltham, MA 02454–9110.

*Correspondence to: Lars Ulrik Wahlberg, CytoTherapeutics, Inc., 701 George Washington Highway, Lincoln, RI 02865.

E-mail: lwahlberg@cyto.com

set of samples. When applied, the ANN uses the input-output mapping learned from the set of training samples to generalize to data not ''seen'' before. The capacity for generalization predominantely depends on network structure (4–7).

In this study, we have constructed an ANN automated cell counting system based on a modified NIH Image software running on a Power Macintosh desktop computer. The system is inexpensive, fast, and reasonably easy to use, but requires some basic knowledge about Pascal programming and ANNs. The ANN functions as an intelligent filter that is used to scan regions of interest in the input image to produce an output image in which debris and other objects classified as ''non-cells'' have been rejected. The output image is then processed using standard image analysis tools available in NIH Image and an object count for the image is thus obtained. The optimal number of hidden neurons of the ANN was evaluated. The system was validated by comparison with blinded human counts, and the performance of the ANN was estimated.

## MATERIALS AND METHODS

### Histology Preparation, System Setup, and Image Acquisition

Cells were encapsulated in a semipermeable and immunoprotective polymer for the purpose of central nervous system drug delivery (1–3). Encapsulated devices, measuring 5 mm in length and 1 mm in diameter, were retrieved from animals (sheep) and in vitro cohorts. Devices were submitted to histology, fixed with 4% paraformaldehyde (Fisher Scientific, catalog no. T353, Pittsburgh, PA), and embedded in glycomethacrylate (GMA, Historesin embedding kit, catalog no. 702218500, Leica, Deerfield, IL), or in paraffin. GMA-embedded devices were sectioned at 4 µm on a Jung supercut 2065 microtome, and paraffin blocks were sectioned at 5 µm on a Reichert-Jung 2030 microtome. Sections were stained with Hematoxylin (Gill's formula, catalog no. CS 402-ID, Fisher Scientific) and Eosin (Eosin Y, catalog no. 17269, Polysciences, Warrington, PA). Slides were then dehydrated and coverslipped using mounting medium (Micromount, catalog no. EM-400, Surgipath, Grayslake, IL).

Microscopy fields were digitized at 50x or 100x magnification (20x or 40x objective, 2.5x video magnification) using an Olympus IMT-2 inverted microscope (Olympus America, Inc., Melville, NY) and a Power Macintosh 7300/180 (Apple Computer, Inc., Cupertino, CA) equipped with the Scion LG-3 frame grabber (Scion Corporation, Frederick, MD). A microscopy stage motor controller, H128AV3 (Prior Scientific, Inc., Rockland, MA), was also included in the system. The H128AV3 controller could move the stage of the microscope in three dimensions, was equipped with an autofocus function, and could be controlled by the computer through an RS232 serial interface.

Sample images used for training of the ANN were manually selected from the digitized microscopy fields and a text file containing the desired outputs was created (see Fig.1A; Fig.2 illustrates a few typical training examples). To this end, a macro was written for the public domain NIH Image program (v. 1.61, developed at the US National Institutes of Health and available on the Internet at http://rsb.info.nih.gov/nih-image/). Only those samples with cells centered in the image were manually tagged with the desired output ''cell-positive.'' Positive and negative samples were denoted +0.9 and -0.9, respectively, because the desired response must be in the output range of the activation function tanh(x) (see section below). The sample sets, consisting of 1,830 images for both magnifications, were compiled into single binary files using a short Pascal program to speed up the subsequent training procedure (Fig. 1B).

### Artificial Neural Network Structure

The network structure was designed to obtain maximal flexibility by allowing the size of the network to be adjusted by changing only a couple of parameters. Three layers were used: two hidden layers, and one output layer (Fig. 3). Furthermore, some a priori knowledge about the samples in the training set was taken into consideration when choosing the structure. The quadrants of the input shared the same weights, and therefore the quadrants were rotated and mirrored so that a cell centered in the input image would give rise to similar input in each quadrant (illustrated by the arcs in Fig. 3). The dimensionality of the input window (the ''retina'') could also be adjusted. Typically, two sizes were employed: 48 by 48 pixels at 100x magnification, and 24 by 24 pixels at 50x magnification; the size of the retina was adjusted according to the size of a cell at these magnifications. The hyperbolical tangent function, tanh(x), was used as the activation function throughout the net. All neurons had a bias weight (negative threshold or ''θ'') in addition to the inputs illustrated in Figure 3; without bias weights some representations cannot be established (4, 10, 11).

### ANN Training

The ANN software consisted of two programs written in Pascal. One was a stand-alone, text-based ANN training program (Fig. 1C) and the other was an ANN-based cell counting system built on the NIH Image software (Fig. 1D). All Pascal code was written and compiled using the Metrowerks CodeWarrior Pro1 Integrated Development Environment (Metrowerks Corporation, Austin, TX). Hence, the parallel ANN was simulated on a serial computer.

The ANN training program (Fig. 1C) contained routines to handle training data, to train the ANN, to store the weights, and to analyze the network performance on the test set (see end of section). The training was initiated by setting the weights to small, normally-distributed, random values with zero mean. The sample set was divided into a training and a test set, the latter of which was used to test the generalization properties of the ANN. The samples were rescaled without normalization so that the pixel intensities fit on the interval [-1,1] because the input data points should not be too close to the flat regions of the tanh(x) function. A sample was automatically picked at

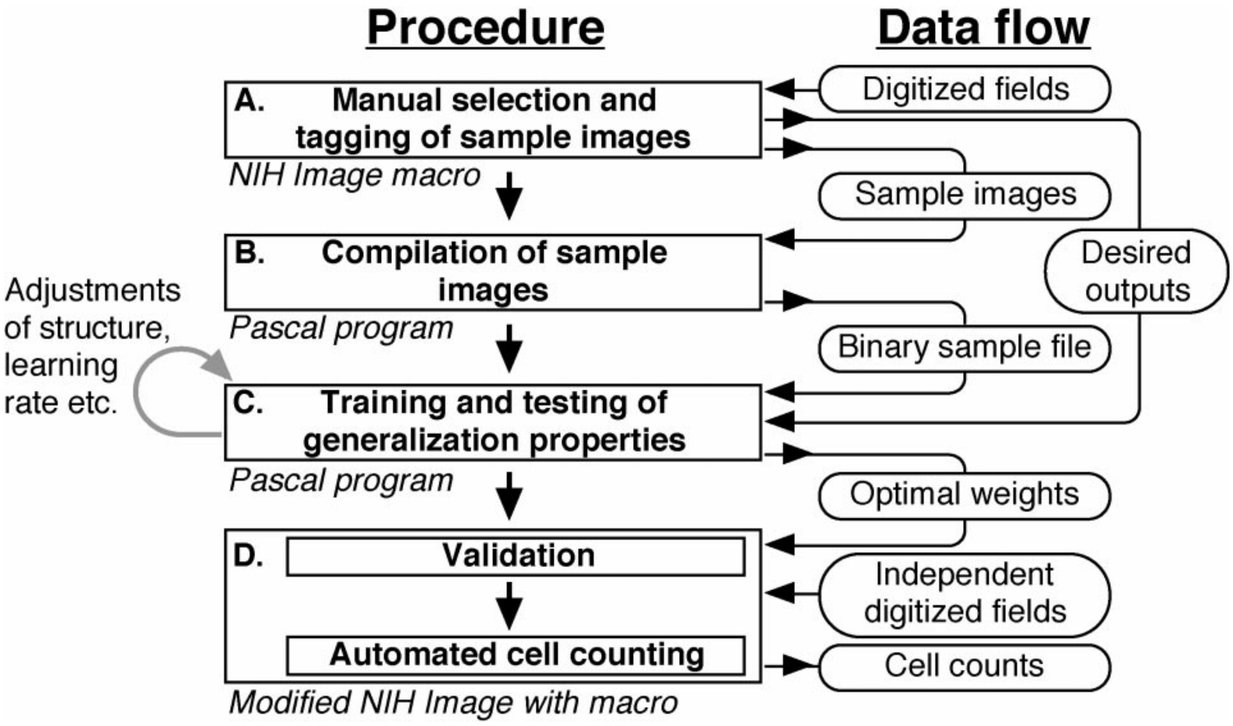
Procedure
Data flow
A. Manual selection and tagging of sample images
NIH Image macro
B. Compilation of sample images
Pascal program
C. Training and testing of generalization properties
Pascal program
D. Validation
Automated cell counting
Modified NIH Image with macro
Adjustments of structure, learning rate etc.
Digitized fields
Sample images
Desired outputs
Binary sample file
Optimal weights
Independent digitized fields
Cell counts

FIG. 1. A schematic illustration of different programs used to select samples and to train, test, validate, and apply the system, as described in Materials and Methods. The data flow is illustrated at the right in the figure. **A:** Sample images were simultaneously user-selected from digitized microscopy fields and manually tagged using an NIH Image macro. **B:** The image samples were compiled into a single binary file using a simple Pascal program to speed up the learning procedure. **C:** The ANN was trained in a stand-alone Pascal program using the samples and the desired outputs generated in A and B. The ANN was subsequently tested for its generalization properties. When necessary, the network stucture and parameters such as the learning rate were adjusted at this point, and the step was repeated. **D:** The optimal weights and the network structure obtained in the previous step were validated using independent microscopy fields. A macro routine running under the modified NIH Image was employed at this step. The software was modified by the addition of ANN-specific UserCode macro instructions (see Materials and Methods). At this point, the system was ready to use for automated cell counting.

random from the training set and was mirrored, flipped upside-down and/or rotated 90° randomly to provide some rotational invariance and also to generate more samples from the available training set. The weights were adjusted using the error back-propagation algorithm and shared weights were updated as described for recurrent nets by Rumelhart et al. (5). This meant that the changes calculated for each ''weight position'' were summed up and used to change the common, shared weight. At least 200,000 iterations were run per training session, which took 1 to 2 h on the Power Macintosh 7300/180, depending on the network structure. This number of iterations was more than sufficient to reach the error minimum for the test set (see end of section). The learning rate parameter was typically initially set to 0.01, but was adjusted periodically so that the learning rate was decreased if the error diminished and vice versa, so-called adaptive learning rate (4, 7). No moment term was used, but batch-learning was used in some cases. The percentage of incorrect classifications and the error variance with respect to the desired output was calculated both for the training and test sets at fixed intervals (typically every 800 iteration) and the weights were stored at the same time. The errors vs. the iteration number were subsequently plotted and analyzed. The weights stored at the error minimum for the test set were selected as the optimal weights, since these give the best generalization properties (4, 7, 11). It was verified that the size of the training set was large enough by examining the generalization properties as the number of samples of the training set was varied. Furthermore, due to the stochastic nature of the learning algorithm and to avoid the ANN getting trapped in a local error minimum (e.g., 4), every training session was repeated five times and the best weights of the five sessions were used.

### Estimating the Optimal Number of Hidden Neurons

The optimal number of neurons in the two hidden layers was estimated as previously described (6). This was performed by changing the number of hidden neurons in the first layer between 2 and 14 and in the second layer between 1 and 8, and evaluating the generalization properties of the ANN on the test set (described above). The procedure was repeated five times for each number of hidden neurons.

### Cell Counting Using the Trained ANN

The modified NIH Image software included a small number of added macro instructions, based on the UserCode call. For example, the actual ANN analysis code was contained in such a macro instruction. The input image of the ANN was analyzed—''scanned''—with the retina of

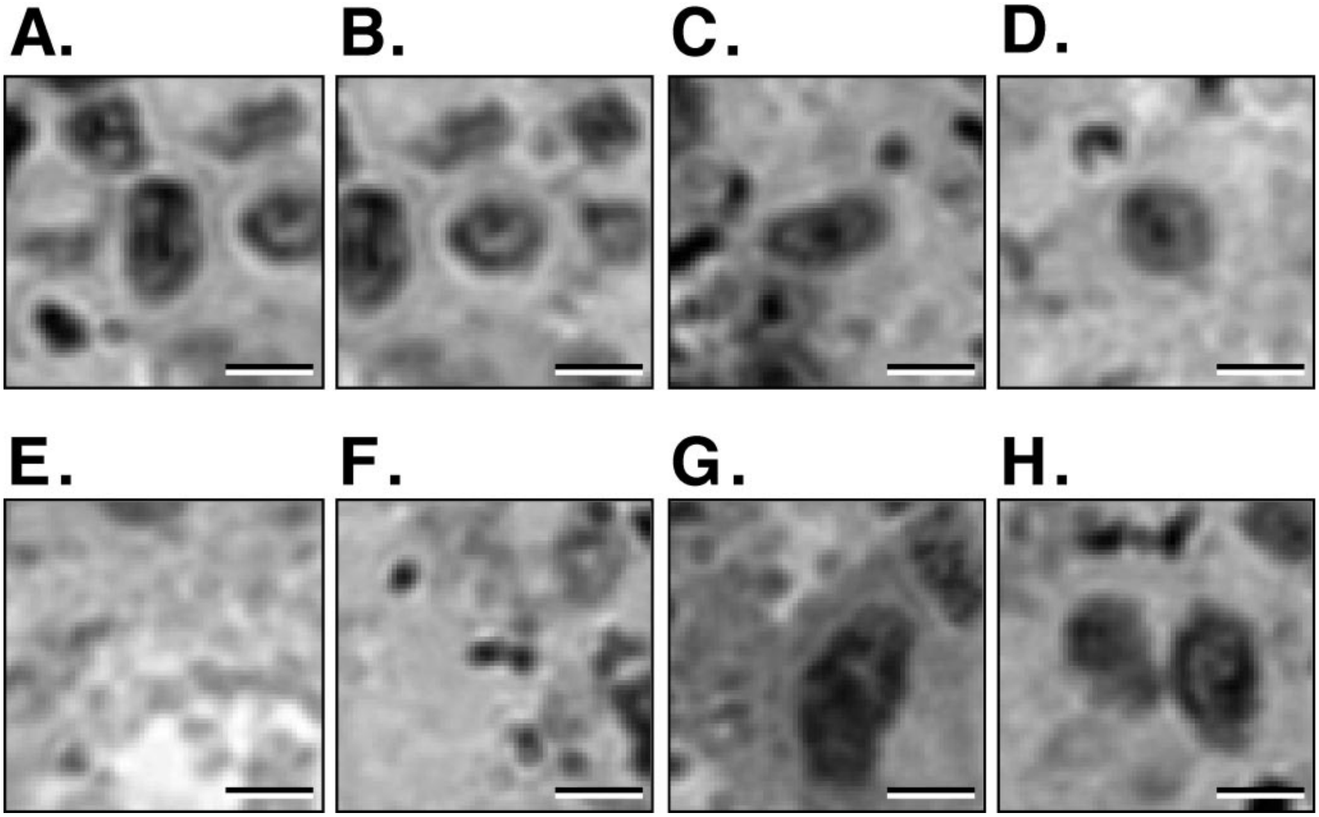

FIG. 2. A few typical examples are illustrated. These are sized 64 by 64 pixels, 8 bits, have 256 gray-scale intensities, and were obtained at a total magnification of 100× (see Materials and Methods). **A–D** contain centered cells and were manually tagged ''cell-positive.'' This means that the corresponding desired output is +0.9. **E–H** do not contain centered cells. Image E is blank, F contains debris, G shows a cell that is not centered, and in image H, the center of the image is located in between the two cells. They are ''cell-negative'' and were manually tagged −0.9. The actual image samples used for training were usually smaller than 64 by 64 pixels: typically 48 by 48 pixels at 100 times magnification and 24 by 24 pixels at 50 times magnification. These pixel areas corresponded to the size of the cells at the two magnifications. Bar = approximately 6 μm and was not present during the ANN training.

the ANN. The feed-forward calculations were performed at each position and an output answer was generated. To speed up the processing, only positions with pixel intensities above a user-defined value were analyzed. The output, ranging from -1 to +1, was rescaled on the interval [0,255] to generate a pixel at the corresponding pixel coordinates in the 8-bit output image. The resulting image was then analyzed with the built-in functions of NIH Image. The output image was converted to a binary image by thresholding at the gray-scale value 128, which corresponded to zero output of the ANN. Connected objects in the binary image were separated using the ultimate eroded points algorithm or watershed segmentation (12, 13). Using the particle analysis function of the NIH Image software, the number of cells, as well as their coordinates, could be ascertained. Usually, a lower object area threshold of not more than 5 pixels was employed to eliminate noise in the image caused by spurious positives.

An NIH Image macro program was written that included the above-mentioned user-defined macro instructions along with code to communicate with the H128AV3 controller (Fig. 1D). This system can automatically digitize microscopy fields, analyze them using the ANN, label the counted cells in the digitized images with crosses, store the labeled images for the ensuing human inspection, and create a text file with the cell numbers obtained for each digitized field.

### Validation Studies

Microscopy fields of undamaged sections (i.e., containing no folds or large artifactual stains) were digitized. The fields were of devices containing the same type of cells and the same type of embedding (paraffin or GMA) as used when training the ANN. The fields were taken from independent sections, i.e., from sections not used for training. Counts were performed both manually and automatically on 20 fields. The manual counting was blinded and performed twice, to account for any bias and to obtain an estimate of person-to-person variability. Validation was performed at both 50x and at 100x magnification.

## RESULTS

### ANN Generalization

Figure 4 illustrates the generalization properties of the ANN for different numbers of neurons in the first layer when trained on 48-by-48-pixel images obtained at 100x magnification, while keeping the size of the second hidden

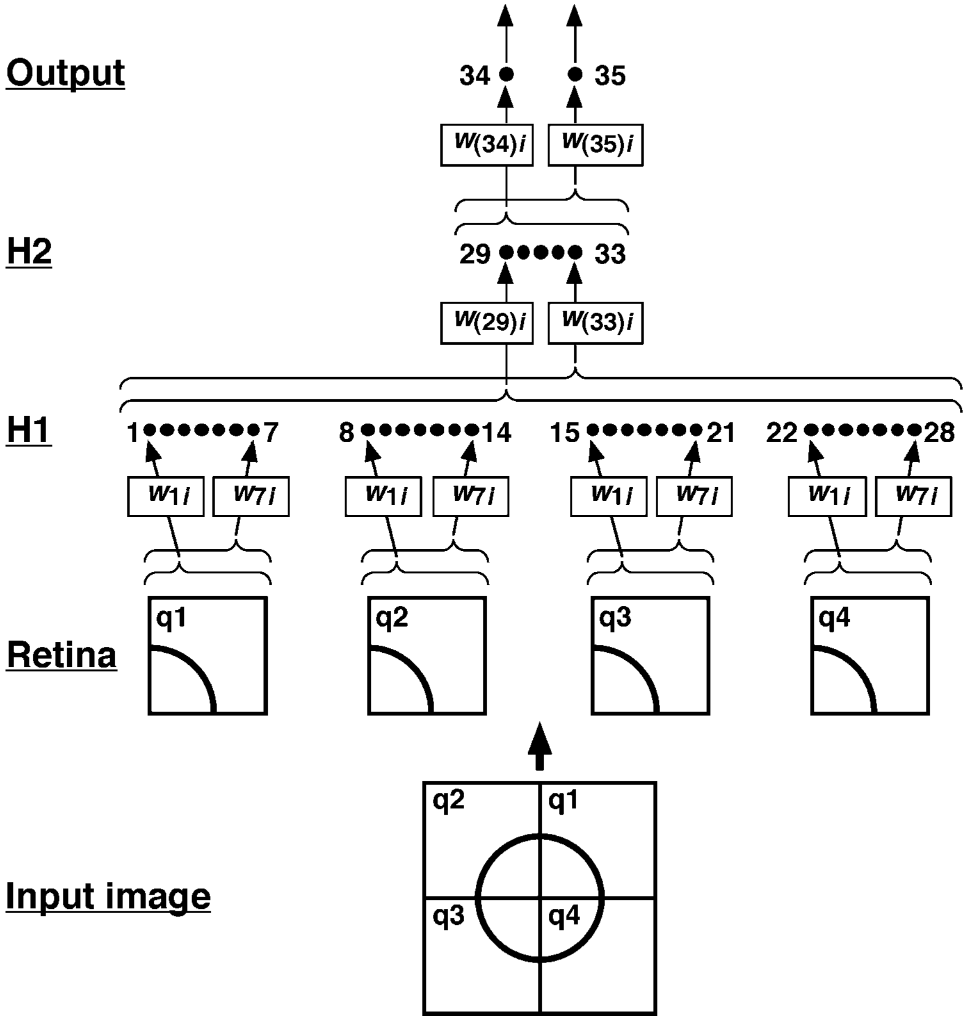

FIG. 3. ANN structure for a 24-by-24-pixel input image (the ''retina''), with 28 neurons in the first and 5 neurons in the second hidden layer. For the sake of clarity, only a few connections are shown. The weighted connection from neuron (or input) $i$ to neuron $j$ is denoted $w_{ji}$. Labels q1 to q4 denote the quadrants of the input image, which are reoriented so that the input view is as similar as possible. Due to the weight sharing, the reorientation enables the ANN to learn a representation easier. The reorientation is illustrated by the arcs, which depict an idealized cell centered in the original input image. A quadrant is fed to seven neurons in the first hidden layer, H1. Weights $w_{1i}$ to $w_{7I}$ are used for neurons 1 through 7, where $i$ goes from 1 to 144, (There are 12 by 12 pixels in a quadrant). This is repeated for the three remaining quadrants using the same weights (weight sharing), thus producing outputs for neurons 8 through 14, 15 through 21, and 22 through 28. The five neurons in the second hidden layer (H2) take the output from all neurons in layer one (H1) as input. For example, neuron 29 receives data from neurons 1 through 28 weighted by $w_{(29)i}$; $i = 1 \ldots 28$. The two output neurons (Output) are fed from the five neurons in the second layer (H2). One output per class is needed (9), hence the two output neurons. These correspond to the ''cell'' and ''non-cell'' classes. Only one output neuron is used during image analysis. All neurons throughout the network have a bias weight in addition to the ilustrated inputs.

layer constant at four neurons. The error variance (the variance of the difference between the actual output and the desired output for the samples in the test set) is plotted vs. the number of neurons. Both the average of five runs and the smallest value of the five runs are graphed. The generalization improved as the number of hidden neurons was increased, but leveled out at around the value of 32 hidden neurons. Therefore, for the first hidden layer, a pick of 32 neurons gave optimal ANN generalization properties. Training became prohibitively slow after 64 hidden neurons and further examination of the generalization properties beyond this number of hidden neurons was not performed. For the second hidden layer, the generalization properties were optimal when 4 neurons were used (not shown).

Similar results were obtained when training the ANN on 24-by-24-pixel images. A choice of 28 and 5 neurons in the first and the second hidden layers was optimal (data not shown).

### Microscopy Field Analysis

The trained ANN is applied on a digitized microscopy field to generate an ''intelligently filtered,'' gray-scale output image, as described in Materials and Methods. A typical digitized field is found in Figure 5A. The results of ANN filtering are shown in Figure 5B. This image is converted to a binary image through thresholding, which is segmented using the watershed algorithm of NIH Image (not shown). The separated objects are counted using the built-in particle analysis function of NIH Image and their

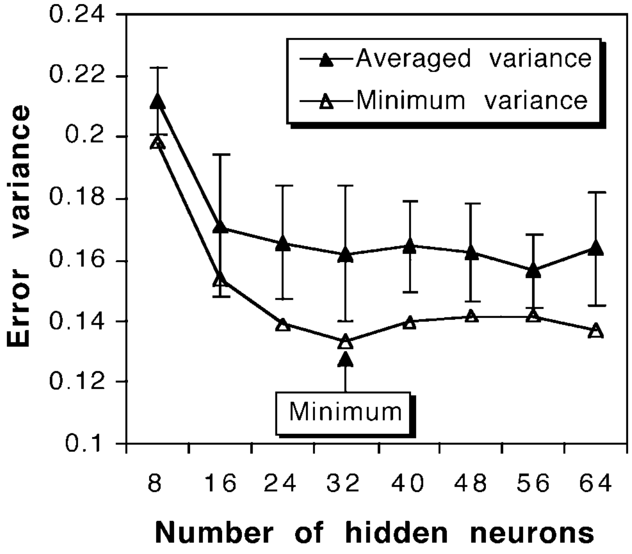

FIG. 4. The effect on the generalization properties of the ANN when changing the number of neurons in the first hidden layer in the case of a 48-by-48-pixel retina. The size of the second hidden layer was kept constant at four neurons, while the number of neurons in the first hidden layer was varied. The ANN variance was averaged over five training sessions. Both the average variance and the minimum variance of the five sessions are plotted. The minimum of the graph indicates the optimal size of the first hidden layer as chosen by the authors.

coordinates are stored. Figure 5C illustrates a superposition of the counts obtained manually (filled circles, by B.R.F.) and using the ANN (crosses). Forty-eight cells were counted manually, while 49 cells were found by the computer. Even when optimizing, direct application of the built-in image analysis methods of NIH Image on the image in Figure 5A gave more than 100 objects (data not shown). Note that even though the numbers of cells counted manually and automatically were very close, the same cells were not always counted (Fig. 5C).

### Validation Studies

To substantiate the ANN counts, these were compared to manual counts by two people. Figure 6 illustrates the results of the validation studies. The blinded counts of P.J.S. and of B.R.F. are plotted vs. the counts of the ANN in 20 digitized microscopy fields. P.J.S. trained the ANN and B.R.F. has expertise in histology, which makes both comparisons important. The correlation indices, the slopes and the intercepts from the validation studies are compiled in Table 1.

### Time Budget

Depending on the size of the ANN and on the amount of information in the image, the analysis of one digitized microscopy field (640 by 480 pixels, 8 bits, 48-by-48-pixel retina) takes between 10 and 40 s on the Power Macintosh 7300/180; 25 s on average is needed for the complete analysis of one digitized field. The ANN with a 24-by-24-pixel retina was faster, requiring approximately a quarter of the multiplications of its larger counterpart and hence a quarter of the processing time. (Note that the area covered at the lower magnification is four times as big as well, making the difference in actual performance 16 times in total.) The manual counting of all the cells in an image takes approximately 3 min, depending on the magnification and on the number of cells present.

## DISCUSSION

NIH Image has been used for cell counting before but, to the authors' knowledge, only in a semi-automated fashion (14–16). With the noisy histological data that the authors are working on, it seems likely that completely automated, practicable cell counting will not be achieved using traditional image analysis software such as NIH Image. A particular cell type may assume different appearances depending on the growth conditions, variability of the staining procedure, illumination during microscopy, etc. Also, the background (cellular debris, polymer matrix, etc.) influences the performance of an automated counter. Simple histogram thresholding and/or gray-scale intensity peak search did not work well with hematoxylin-eosin histology, because the cells in digitized microcopy fields may have more than one intensity peak each, or they may appear as disconnected objects when thresholding (not shown). The aim of this study was to apply artificial intelligence in the form of an ANN to create an automated cell counter with augmented accuracy as compared to traditional image analysis tools, and increased speed in comparison with manual counts.

There were two basic reasons why an ANN technique was employed. First, ANNs learn by example (12, 4, 6, 7). Hence, no rules need to be defined to program an ANN. This was ideal for our study, since it is difficult to describe explicitly the appearance of a cell in a digitized microscopy field in terms of parameters based on gray-scale intensities, shape after thresholding, etc. Furthermore, learning by example implies that human expertise may be mimicked. Second, as opposed to statistical methods, ANNs are non-linear and may perform better than their statistical, linear counterparts (10). This means that they may identify relationships between variables not previously appreciated (8). However, statistical methods and ANNs are essentially identical in many respects (6).

In this study, the image data was fed directly to the ANN. Others have approached the problem differently. One method has been reported that entails the extraction of a number of features from the image or a region of interest of the image. These features are subsequently used to train the ANN (9, 17). The subsequent ANN analysis involves extracting the same features from images to be analyzed and thus letting the ANN classify the image data indirectly. Another method is to use pixel samples in the region of interest (18). Both methods are aimed at reducing the number of inputs to the ANN, which, in turn, reduces the size of the ANN and the amount of computation involved. In addition, large nets have more degrees of freedom, which requires larger training sets (4). In our study, the degrees of freedom were minimized by estimating the number of hidden neurons required for optimal generaliza-

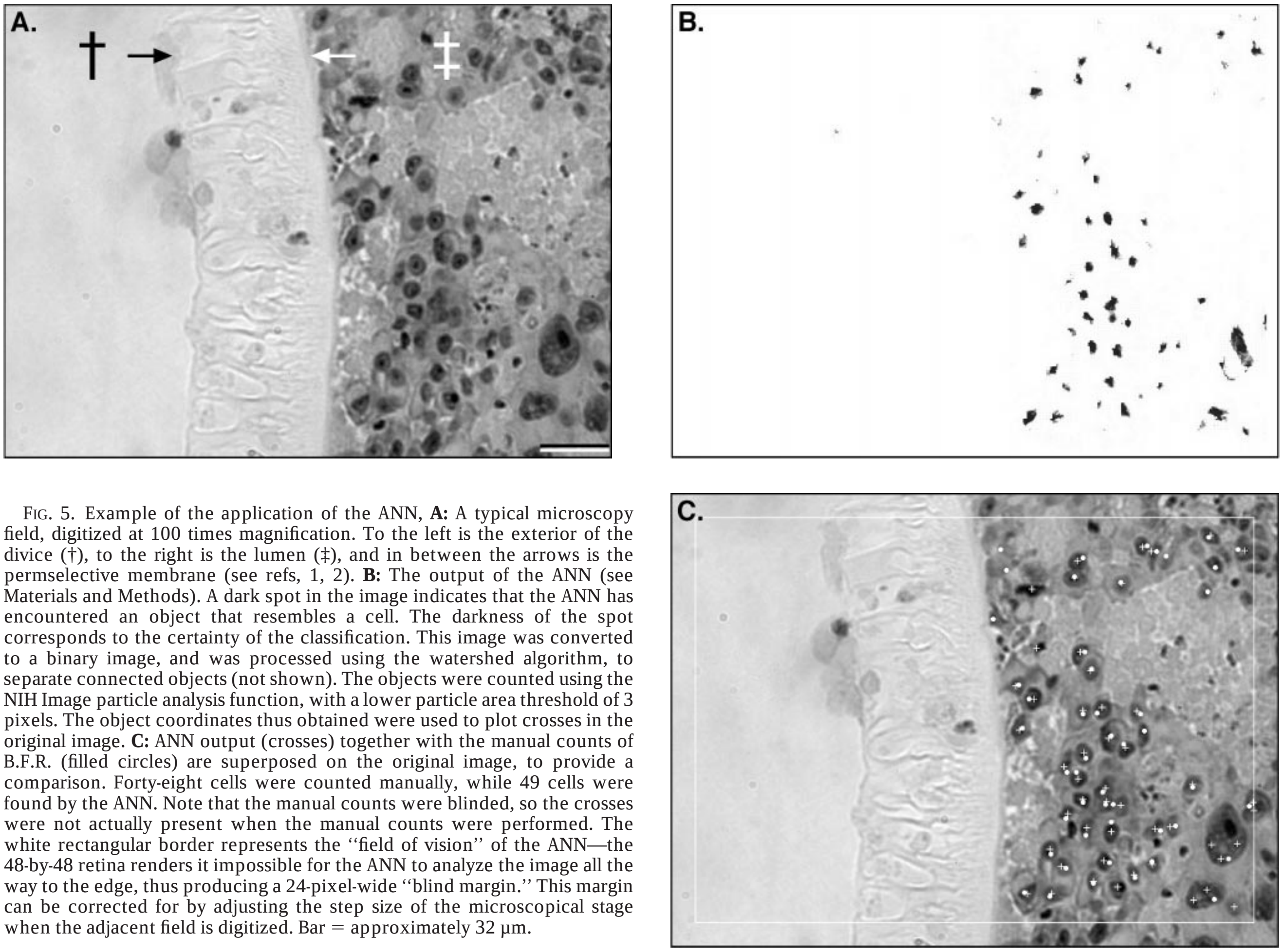

FIG. 5. Example of the application of the ANN, **A:** A typical microscopy field, digitized at 100 times magnification. To the left is the exterior of the divice (†), to the right is the lumen (‡), and in between the arrows is the permselective membrane (see refs, 1, 2). **B:** The output of the ANN (see Materials and Methods). A dark spot in the image indicates that the ANN has encountered an object that resembles a cell. The darkness of the spot corresponds to the certainty of the classification. This image was converted to a binary image, and was processed using the watershed algorithm, to separate connected objects (not shown). The objects were counted using the NIH Image particle analysis function, with a lower particle area threshold of 3 pixels. The object coordinates thus obtained were used to plot crosses in the original image. **C:** ANN output (crosses) together with the manual counts of B.F.R. (filled circles) are superposed on the original image, to provide a comparison. Forty-eight cells were counted manually, while 49 cells were found by the ANN. Note that the manual counts were blinded, so the crosses were not actually present when the manual counts were performed. The white rectangular border represents the ''field of vision'' of the ANN—the 48-by-48 retina renders it impossible for the ANN to analyze the image all the way to the edge, thus producing a 24-pixel-wide ''blind margin.'' This margin can be corrected for by adjusting the step size of the microscopical stage when the adjacent field is digitized. Bar = approximately 32 µm.

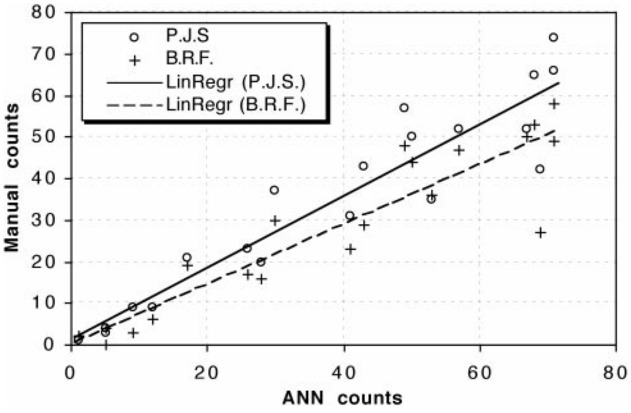

FIG. 6. Results of the validation studies. Twenty microscopy fields digitized at 100 times magnification were counted using the ANN (see Materials and Methods) and twice manually by two people (P.J.S. and B.F.R.). The lines were obtained using linear regression (see Table 1).

Table 1
*Regression Data Obtained From Validation Studies**

| Magnification | Regression variables | Correlation index ($R^2$) | Slope | Intercept |
|---|---|---|---|---|
| 100× | P.J.S. vs. ANN | 0.8829 | 0.8673 | 1.2214 |
| | B.R.F. vs. ANN | 0.8487 | 0.7193 | 0.2866 |
| | B.R.F. vs. P.J.S. | 0.9402 | 0.8202 | −0.4095 |
| 50× | P.J.S. vs. ANN | 0.6927 | 0.8145 | 4.0931 |
| | B.R.F. vs. ANN | 0.6214 | 0.9313 | 1.7512 |
| | B.R.F. vs. P.J.S. | 0.9772 | 1.1934 | −7.0205 |

*Twenty microscopy fields digitized at 100 and 50 times magnification were counted using the ANN (see Materials and Methods) and twice manually by two people (P.J.S. and B.F.R.). The slope and the intercept refer to the linear regression curve (also see Fig. 6).

tion properties. Also, the extensive weight sharing in the first hidden layer reduced the degrees of freedom by one fourth. In addition, the weights at the error minimum on the test set—the optimal weights—were selected to avoid overfitting due to possible excess in degrees of freedom and thus to ensure a practical level of generalization (6, 11). In either case, whether a large ANN is used directly on the image data, or image features are extracted prior to the analysis with a smaller ANN, a fast computer is needed for the computations. It is worth noting that in the latter case, the fewer degrees of freedom of a smaller ANN leads to a smaller number of required examples for learning. This is a possible solution to the above-mentioned problem of prohibitively large data sets needed for training large ANNs.

Even though the ANN was good at distinguishing cells from debris in the images, it was more difficult to train the network to classify different types of cells based on their morphology (data not shown). It appears that a more fruitful approach for cell classification would be to have several ANNs, each trained for a different task. Cells could be identified using one network, while the subsequent classification of the cells could be performed by another neural net. Similar approaches have been described elsewhere in the literature (e.g., 18, 19).

When compared to a human, it is apparent that the ANN program makes mistakes (Fig. 5C). However, unlike the method employing only the standard NIH Image algorithms, the ANN produces considerably fewer and more consistent errors. The error rate is comparable to the person-to-person variability (Table 1). Thus, this ANN program has practical use. No systematic errors or nonlinearities in the ANN performance were found (Fig. 6). The validation data indicates that the ANN performance was more similar to P.J.S. than to B.R.F. This may be due to mimicking; P.J.S. selected the samples for the ANN training set.

A few problems remain to be resolved:

- Many of the pitfalls of the system are related to inadequate histology. The importance of good histological preparation for computer-assisted analysis has been reported previously (e.g., 20). Sections fold easily and it may be difficult to achieve consistent staining. The ANN used in this study produced erratic outputs when folds or large dark spots were encountered (outliers in Fig. 6). The ANN can be triggered by the edges of or irregularities in these dark areas, and the counts may thus be increased. To avoid these problems, it should be possible to train an additional ANN to recognize folds and other imperfections in digitized fields, so that either the image may be discarded or the area containing the imperfection may be avoided. The removal of the worst outlier in Figure 6 augments the correlation index to $R^2 = 0.92$, which is close to person-to-person variability (Table 1). Most problems due to artifactual histology may also be avoided by visual inspection prior to or following the ANN analysis.
- In our hands, a final magnification of 100x worked well and yielded better data than at 50x (Table 1). Higher magnifications could most likely further improve performance. However, this would require a larger ANN and more digitized fields to be analyzed for the same area, which in turn would significantly slow down the overall operation of the system.
- A large amount of training examples is needed for good generalization properties. A method to rapidly generate several examples from a few available ones would benefit the ANN approach to automated cell counting.
- Standard desktop computers are today powerful enough to be used with ANNs, but the lack of computing power makes the software less flexible and user-friendly than desired. With even faster and more sophisticated computers (faster processors, parallel processors, ANN hardware, etc.), these problems should be alleviated and features such as window handling and ANN flexibility may be integrated.

It should be pointed out that there are other approaches to achieve high-performance automated object counting in noisy images. Principal component analysis is frequently employed (e.g., 1), which is a linear and statistical method. In the case of histological data, the staining method should ideally be selected for the sole purpose of cell counting (e.g., a nuclear stain). Unfortunately, the histological method is often chosen with several purposes in mind, not only cell counting.

In conclusion, this study has shown that the construction of an automated cell counter based on ANN technology is feasible. There are several benefits to this system, such as speed of analysis, consistency, and the automation in itself. The ANN-based counter is considerably faster than a human (approximately six times faster), and is likely to get faster as computer technology improves. The automated counter is consistent—it invariably produces the same counts for a given digitized microscopy field, something that is not true for its human counterpart. Finally, manual counting is tedious, inexact, time-consuming, and expensive. Freeing personnel for other, more demanding tasks is an important technological advancement in the histological laboratory.

## ACKNOWLEDGMENTS

The authors acknowledge the assistance with artificial neural networks provided by Dr. Mats Gustafsson (Signals and Systems Group, Uppsala University, Sweden) as well as the helpful information on image analysis methods provided by Dr. Ingela Nyström (Centre for Image Analysis, Uppsala University, Sweden).

## LITERATURE CITED

1. Aebischer P, Wahlberg L, Tresco PA, Winn SR. Macroencapsulation of dopamine-secreting cells by coextrusion with an organic polymer solution. Biomaterials 1991;12:50–56.
2. Flanagan TR, Frydel B, Tente B, Lavoie M, Doherty E, Rein D, Emerich DF, Winn SR. Test for validating the safety of encapsulated xenografts. In: Flanagan TR, Emerich DF, Winn SR, editors. Providing pharmacological access to the brain: alternate approaches. San Diego: Academic Press; 1994. p 403–423.
3. Winn SR, Tresco PA. Hydrogel applications for encapsulated cellular transplants. In: Flanagan TR, Emerich DF, Winn SR, editors. Providing pharmacological access to the brain: alternate approaches. San Diego: Academic Press; 1994. p 387–402.
4. Haykin S. Neural networks: a comprehensive foundation. New York: Macmillan College Publishing Company, Inc.;1994.
5. Rumelhart DE, Hinton GE, Williams RJ. Learning internal representations by error propagation. In: Rumelhart DE, McClelland JL, editors. Parallel distributed processing: explorations in the microstructure of cognition. Cambridge, MA: MIT Press; 1986. p 318–362.
6. Swingler K. Applying neural networks: a practical guide. London: Academic Press; 1996.
7. Tsoukalas LH, Uhrig RE. Fuzzy and neural approaches in engineering. New York: Wiley-Interscience; 1997.
8. Baxt WG. Use of an artificial neural network for the diagnosis of myocardial infarction. Ann Intern Med 1991;115:843–848.

9. Brouwer RK, MacAuley C. Classifying cervical cells using a recurrent neural network by building basins of attraction. Anal Quant Cytol Histol 1995;17:197–203.
10. Kattan MW, Beck JR. Artificial neural networks for medical classification decisions. Arc Pathol Lab Med 1995;119:672–677.
11. Tu JV. Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes. J Clin Epidemiol 1996;49:1225–1231.
12. Gonzalez RC, Woods RE. Digital image processing. Reading, MA: Addison-Wesley Publishing Company; 1993.
13. Russ JC. The image processing handbook. Boca Raton, FL: CRC Press; 1995.
14. Furness PN, Rogers-Wheatley L, Harris KP. Semiautomatic quantitation of macrophages in human renal biopsy specimens in proteinuric states. J Clin Pathol 1997;50:118–122.
15. Gatlin CL, Schaberg ES, Jordan WH, Kuyatt BL, Smith WC. Point counting on the Macintosh. A semiautomated image analysis technique. Anal Quant Cytol Histol 1993;15:345–350.
16. Ide M, Jimbo M, Yamamoto M, Kubo O. Tumor cell counting using an image analysis program for MIB-1 immunohistochemistry. Neurol Med Chir (Tokyo) 1997;37:158–162.
17. Kolles H, von Wangenheim A, Rahmel J, Niedermayer I, Feiden W. Data-driven approaches to decision making in automated tumor grading. An example of astrocytoma grading. Anal Quant Cytol Histol 1996;18:298–304.
18. Schaberg ES, Jordan WH, Kuyatt BL. Artificial intelligence in automated classification of rat vaginal smear cells. Anal Quant Cytol Histol 1992;14:446–450.
19. Mehdi B, Stacey D, Harauz G. A hierarchical neural network assembly for classification of cervical cells in automated screening. Anal Cell Pathol 1994;7:171–180.
20. Kruger T. Computer-assisted sperm analysis systems: morphometric aspects. Hum Reprod 1995;10:46–52.
21. Burl MC, Asker L, Smyth P, Fayyad U, Perona P, Crumpler L, Aubele J: Learning to recognize volcanoes on Venus. Mach Learn 1998;30:165–194.